CS771 - Assignment 1 (The Squared SVM Solver)

Nikhil Bansal(170431) Gosai Akash(170278) Vaibhav Jindal(170775)

Pratyush(170500) Shivam Kumar(170669)

1 Lagrangian for P2

Solution: The constraints for P2 are :

$$y^i \langle \mathbf{w}, \mathbf{x}^i \rangle \geq 1 - \xi_i \text{ for all } i \in [n]$$
$$\implies 1 - \xi_i - y^i \langle \mathbf{w}, \mathbf{x}^i \rangle \leq 0 \text{ for all } i \in [n]$$

If we denote the equation just derived as g_i for i ∈ [n], then we will have a dual variable α_i corresponding to g_i for all $i \in [n]$
Thus the Lagrangian for P2 is :

$$L(\mathbf{w}, \boldsymbol{\xi}, \boldsymbol{\alpha}) = \tfrac{1}{2}\|\mathbf{w}\|_2^2 + C \cdot \textstyle\sum_{i=1}^n \xi_i^2 + \sum_{i=1}^n \alpha_i (1 - \xi_i - y^i \langle \mathbf{w}, \mathbf{x}^i \rangle)$$

2 Dual derivation by eliminating primal variables

Solution: The Lagrangian of the primal problem given is:

$$L(\mathbf{w}, \boldsymbol{\xi}, \boldsymbol{\alpha}) = \tfrac{1}{2}\|\mathbf{w}\|_2^2 + C \textstyle\sum_{i=1}^n \xi_i^2 + \sum_{i=1}^n \alpha_i (1 - \xi_i - y^i \langle \mathbf{w}, \mathbf{x}^i \rangle)$$

In the above equation,**w** and $\boldsymbol{\xi}$ are primal variables and $\boldsymbol{\alpha}$ is a dual variable.
Then, the corresponding dual problem would be :

$$\max_{\alpha_i \geq 0} \Big(\min_{\mathbf{w} \in R^d, \boldsymbol{\xi} \in R^n} \Big(L(\boldsymbol{w}, \boldsymbol{\xi}, \boldsymbol{\alpha}) \Big) \Big)$$

Then by the primal optimality condition, for the minimum value of Lagrangian,

$$\frac{\partial L}{\partial \mathbf{w}} = 0 \implies \mathbf{w} = \sum_{i=1}^n \alpha_i y_i \mathbf{x}^i$$
$$\frac{\partial L}{\partial \boldsymbol{\xi}} = 0 \implies \boldsymbol{\xi} = \frac{\boldsymbol{\alpha}}{2C}$$

Then the dual problem can be written as :
D2= $\max_{\alpha_i \geq 0} \Big(\sum_{i=1}^n \alpha_i + \frac{1}{2}\|\mathbf{w}\|^2 + \sum_{i=1}^n \xi_i (C\xi_i - \alpha_i) - \sum_{i=1}^n (\alpha_i y_i \langle \mathbf{x}^i, \mathbf{w} \rangle) \Big)$

Substituting $\mathbf{w} = \sum_{i=1}^n (\alpha_i y_i \mathbf{x}^i)$ and $\xi_i = \frac{\alpha_i}{2C}$
D2= $\max_{\alpha_i \geq 0} \Big(\sum_{i=1}^n \alpha_i + \sum_{i=1}^n \frac{\alpha_i}{2C} (\frac{\alpha_i}{2} - \alpha_i) - \frac{1}{2} \langle \sum_{i=1}^n (\alpha_i y_i \mathbf{x}^i), \sum_{j=1}^n (\alpha_j y_j \mathbf{x}^j) \rangle \Big)$
D2= $\max_{\alpha_i \geq 0} \Big(\sum_{i=1}^n \alpha_i - \sum_{i=1}^n \frac{\alpha_i^2}{4C} - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n (\alpha_i \alpha_j y_i y_j \langle \mathbf{x}^i, \mathbf{x}^j \rangle) \Big)$

D2 is the final dual problem obtained after elimination of primal variables **w** and $\boldsymbol{\xi}$.

3 Methods Implemented

3.1 Mini-batch SGD on P1

Solution: Here the optimization problem P1 is:

$$\min_{\mathbf{w}\in R^d} f(\mathbf{w}), \text{where}$$

$$f(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\Big([1 - y^i\langle\ \mathbf{w}, \mathbf{x}^i\rangle\ ]_+\Big)^2 \tag{1}$$

$$\nabla(f(\mathbf{w})) = \mathbf{w} + 2C\sum_{i=1}^{n}\Big([1 - y^i\langle\ \mathbf{w}, \mathbf{x}^i\rangle\ ]_+ g^i y^i \cdot \mathbf{x}^i\Big) \tag{2}$$

Here , $g^i \in \nabla l_{hinge}\big(y^i \cdot \langle\ \mathbf{w}, \mathbf{x}^i\rangle\ \big)$
For implementing stochastic gradient descent, we chose B random points $(\mathbf{x}^{i_t^1}, y^{i_t^1}), ..., (\mathbf{x}^{i_t^B}, y^{i_t^B})$ and used

$$\nabla(f(\mathbf{w})) \approx \mathbf{w} + 2C\frac{n}{B}\sum_{b=1}^{B}\Big([1 - y^{i_t^b}\langle\ \mathbf{w}, \mathbf{x}^{i_t^b}\rangle\ ]_+ g^{i_t^b} y^{i_t^b} \cdot \mathbf{x}^{i_t^b}\Big) \tag{3}$$

We used this value of gradient to update w in each iteration of the solver i.e.

$$\mathbf{w} \leftarrow \mathbf{w} - lr * \nabla(f(\mathbf{w})) \tag{4}$$

In the implementation of the above algorithm, we used the following hyperparameters:
Batch size B = $2^{10} = 1024$
Steplength lr = $\frac{0.0003}{\sqrt{t}}$ (t = iteration count of the loop).

3.2 Coordinate maximisation on D2

Solution: The coordinate maximisation was used on the dual problem D2 given by

$$\text{D2= } \max_{\alpha_i \geq 0}\Big(\sum_{i=1}^{n}\alpha_i - \sum_{i=1}^{n}\frac{\alpha_i^2}{4C} - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}(\alpha_i\alpha_j y_i y_j\langle\mathbf{x}^i, \mathbf{x}^j\rangle)\Big)$$

Differentiating the dual objective with respect to α_i we get:

$$1 - \frac{\alpha_i}{2C} - \sum_{j=1, i\neq j}^{n}(\alpha_j y_j y_i\langle\mathbf{x}^i, \mathbf{x}^j\rangle) - \alpha_i\langle\mathbf{x}^i, \mathbf{x}^i\rangle$$

Putting the derivative equal to 0 and solving we get the value of α_i as:

$$\alpha_i = \frac{1 - \sum_{j=1, i\neq j}^{n}(\alpha_j y_j y_i\langle\mathbf{x}^i, \mathbf{x}^j\rangle)}{\langle\mathbf{x}^i, \mathbf{x}^i\rangle + \frac{1}{2C}}$$

The numerator can be calculated efficiently as:

$$\sum_{i\neq j}\alpha_j y_j\langle\mathbf{x}^i, \mathbf{x}^j\rangle = \langle w, \mathbf{x}^i\rangle - \alpha_i y_i\langle\mathbf{x}^i, \mathbf{x}^i\rangle$$

In every iteration α_i is updated and then **w** is also updated using the equation

$$\mathbf{w}_{new} = \sum_{i=1}^{n}\alpha_i y_i\mathbf{x}^i$$

which can be reduced to

$$\mathbf{w}_{new} = \mathbf{w}_{old} + (\alpha_{new} - \alpha_{old})y_i\mathbf{x}^i$$

The value of i is chosen cyclically and the α_i's are initialised to zero.

3.3 Coordinate descent on P1

Solution: The derivative of the P1 objective with respect to w is given as:

$$\nabla(f(\mathbf{w})) = \mathbf{w} - 2C\sum_{i=1}^{n}\left([1 - y^i\langle\ \mathbf{w}, \mathbf{x}^i\rangle\ ]_+ y^i \cdot \mathbf{x}^i\right) \tag{5}$$

To reduce the time taken for each update, we choose a batch size, $B = 512$ for this method and calculate the gradient for this minibatch. Thus we are using the approximate formula:

$$\nabla(f(\mathbf{w})) \approx \mathbf{w} - 2C\frac{n}{B}\sum_{i=1}^{B}\left([1 - y^i\langle\ \mathbf{w}, \mathbf{x}^i\rangle\ ]_+ y^i \cdot \mathbf{x}^i\right) \tag{6}$$

At every step, we choose a variable weight w_i randomly and calculate the gradient with respect to w_i using the above formula. Note that the gradient is calculated only with respect to w_i and not the entire vector w.

The weight w_i is updated as:

$$w_i = w_i - learning\_rate * \frac{\partial f(w)}{\partial w_i}$$

3.4 Coordinate minimization on P1

Solution: Here for the coordinate minimization in P1, in the gradient of f(**w**) i.e.

$$\nabla(f(\mathbf{w})) = \mathbf{w} - 2C\sum_{j=1}^{n}\left([1 - y^j\langle\ \mathbf{w}, \mathbf{x}^j\rangle\ ]_+ y^j \cdot \mathbf{x}^j\right) \tag{7}$$

Consider any i ∈ [n]. Then,

$$\frac{\partial f(\mathbf{w})}{\partial w_i} = w_i - 2C\sum_{j=1}^{n}\left([1 - y^j\langle\ \mathbf{w}, \mathbf{x}^j\rangle\ ]_+ y^j \cdot x_i^j\right) \tag{8}$$

Let k ∈ [n] s.t. $1 - y^k\langle\ \mathbf{w}, \mathbf{x}^k\rangle\ \geq 0$.Then,

$$\frac{\partial f(\mathbf{w})}{\partial w_i} = w_i - 2C\sum_{k}\left((1 - y^k\langle\ \mathbf{w}, \mathbf{x}^k\rangle\ )y^k \cdot x_i^k\right) \tag{9}$$

Putting $\frac{\partial f(\mathbf{w})}{\partial w_i} = 0$, we have :

$$w_i = 2C\sum_{k} y_k x_i^k - \sum_{k} x_i^k\langle\ \mathbf{w}, \mathbf{x}^k\rangle \tag{10}$$

Adding $2C\sum_k (x_i^k)^2 w_i$ both sides, and rearranging we get:

$$w_i = \frac{2C\sum_k y_k x_i^k - \sum_k x_i^k\langle\ \mathbf{w}, \mathbf{x}^k\rangle\ + 2C\sum_k (x_i^k)^2 w_i}{1 + 2C\sum_k (x_i^k)^2 w_i} \tag{11}$$

In every iteration of the loop, we choose i cyclically s.t. $i \in [n]$ and update w_i according to the above equation to minimize the objective function in that direction.

4 Deciding values of hyperparameters

4.1 Mini-batch SGD on P1

The hyperparmeters involved for mini-batch SGD are batch size (B) and step length i.e.learning rate (lr). For deciding the value of B, we preferred only those values which are some power of 2, as it helps in faster operations. So we experimented with B = 2^n where n = 2,3,4,... Finally, 1024 was found to be a suitable value of batchsize.
For the optimum value of learning rate, we first tried with lr = 0.1,0.03,0.01,0.003,0.001 and so on.In this case

the most optimum value was obtained for lr = 0.0003.For the nature of variance of learning rate, we tried $\frac{1}{t}$, $\frac{1}{\sqrt{t}}$ and other exponents of t like 0.3,0.5,0.7 etc. The rate of convergence was found to be fastest for the decay rate of $\frac{1}{\sqrt{t}}$.Hence, learning rate was fixed to be:

$$lr = 0.0003/\sqrt{t}$$

where t is the iteration count.

4.2 Coordinate maximization on D2

The only parameter to decide in this method was how to decide which coordinate to maximise in each iteration. So we experimented two methods of choosing k (coordinate to be maximized) :
a) random value of k
b) choosing k cyclically.
The overall maxima achieved across different trials for both the methods was almost same but the fluctuations were higher for random k. Hence to return precisely better maxima every time, we chose cyclic method to choose k.

4.3 Coordinate descent on P1

We experimented with different values of initial weights. We tried uniform random initialization, gaussian initialization and zero initialization of weights. We found out that setting weights to zero initially gives better convergence.
Also gradient was calculated not using the entire data but batches of data. Different batch-sizes were tried and finally we fixed the value of batch size to 512.
Learning rate was the most important hyper-parameter. We tried several values of initial learning rates and tried both constant and decaying learning rate. For value of learning rate 0.001 and the decay of $\frac{1}{\sqrt{t}}$, the convergence was found to be the fastest. So, learning rate was fixed to be:

$$learning\_rate = \frac{0.001}{\sqrt{t}}$$

where t is the iteration count.
Also we tried choosing the co-ordinates randomly and cyclically. Finally choosing randomly was found to be better.

4.4 Coordinate minimization on P1

The only hyper-parameter involved in the method was the choice of w_i to be updated. We chose the w_i to be updated cyclically. This choice was found to decrease the objective function very quickly and reached the optimum value of objective value in the minimum time compared to other methods.

5 Graph of results obtained

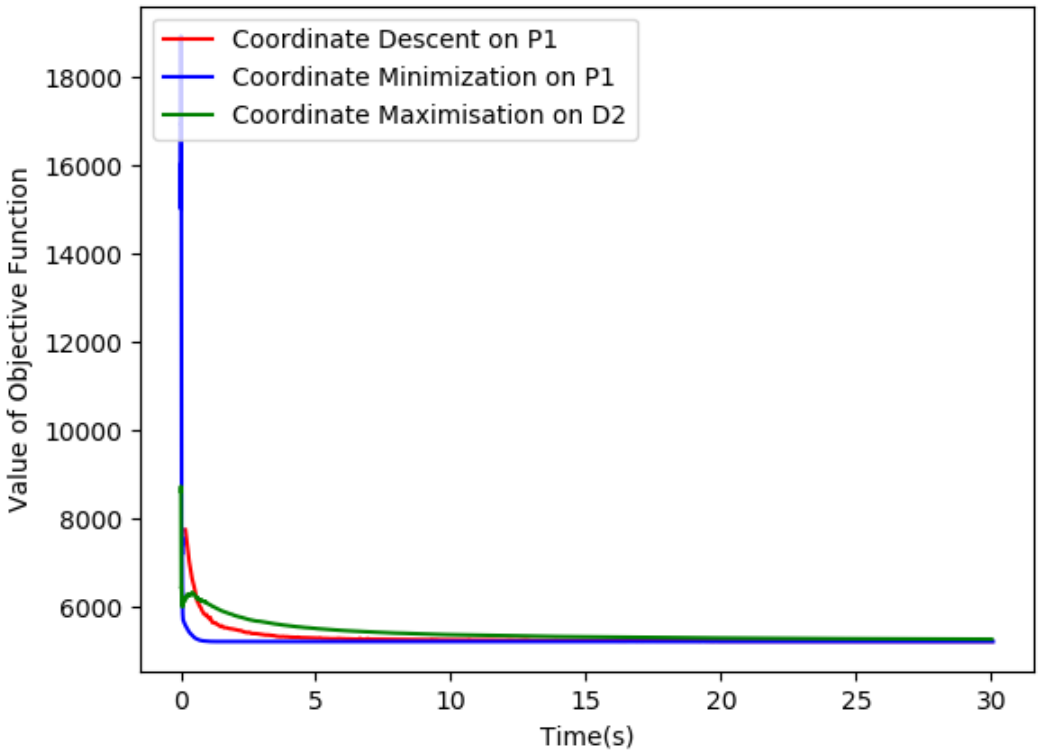
We plot the training loss vs. time graph for three methods:

- Coordinate Descent on P1
- Coordinate Minimization on P1
- Coordinate Maximisation on D2

For each of these plots, we sample a point on the curve after 100 updates to w and b. It was observed that the plot for coordinate maximisation on D2 showed a lot of noise and jittering. To eliminate this problem, we maintained w_{run} and b_{run} variables in this case to smoothly vary the learnable parameters and obtained a smooth curve(green).
Following observations can be made from the given plot:

- Coordinate Minimisation on P1(blue) works best for this dataset and converges to the optimal value within 1 second of the start of the training procedure.

- Coordinate Descent on P1(red) converges at around 6 seconds from the start of the procedure.
- Coordinate Maximisation on D2(green) converges at around 14 seconds. It also shows some random variation towards the beginning of the training but eventually converges smoothly.



6 Code Submission

In our experiments we found that the method of Coordinate Minimization on P1 works best among all the other methods. Training loss was converging to the minimum value using this method and also the loss converges the fastest for this method. Moreover, there are no hyper-parameters in this method and hence we are submitting the code for this particular method. The code can be found here.

7 Bonus Question

Let's consider the problem P2 with extra constraints $\xi_i \geq 0$ for all $i \in [n]$.
The Lagrangian of this problem will be:

$$L(w,\xi,\alpha,\beta) = \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^{n}\xi_i^2 + \sum_{i=1}^{n}\alpha_i(1-\xi_i - y^i\langle \mathbf{w}, \mathbf{x}^i\rangle) - \sum_{i=1}^{n}\beta_i\xi_i$$

In the above equation,$\mathbf{w}$ and ξ are primal variables and α and β are dual variables.
Then, the corresponding dual problem would be :

$$\max_{\alpha_i \geq 0, \beta_i \geq 0} \left(\min_{w \in R^d, \xi \in R^n} \left(L(w,\xi,\alpha,\beta) \right) \right)$$

Then by the primal optimality condition, for the minimum value of Lagrangian,

$$\frac{\partial L}{\partial w} = 0 \implies w = \sum_{i=1}^{n} \alpha_i y_i x_i$$

$$\frac{\partial L}{\partial \xi} = 0 \implies \xi_i = \frac{\alpha_i + \beta_i}{2C}$$

Then the dual problem (let's call it D_2') can be written as :

$$D_2' = \max_{\alpha_i \geq 0, \beta_i \geq 0} \left(\sum_{i=1}^{n} \alpha_i + \frac{1}{2}\|w\|^2 + \sum_{i=1}^{n} \xi_i(C\xi_i - \alpha_i - \beta_i) - \sum_{i=1}^{n}(\alpha_i y_i \langle x_i, w \rangle) \right)$$

Substituting $\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i x_i$ and $\xi_i = \frac{\alpha_i + \beta_i}{2C}$

$$D_2' = \max_{\alpha_i \geq 0, \beta_i \geq 0} \left(\sum_{i=1}^{n} \alpha_i + \sum_{i=1}^{n} \frac{\alpha_i + \beta_i}{2C}(C * \frac{\alpha_i + \beta_i}{2C} - \alpha_i - \beta_i) - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i y_i \alpha_j y_j \langle x_i, x_j \rangle \right)$$

$$D_2' = \max_{\alpha_i \geq 0, \beta_i \geq 0} \left(\sum_{i=1}^{n} \alpha_i - \sum_{i=1}^{n} \frac{(\alpha_i + \beta_i)^2}{4C} - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} (\alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle) \right)$$

Since, the above optimization problem contains β only in one term we can easily calculate the value of β for which the objective is maximized.

Solving $\frac{\partial D_2'}{\partial \beta_i}$ = 0, we get:

$$-(\alpha_i + \beta_i)/2C = 0$$

This implies $\alpha_i + \beta_i = 0$

So, $\beta_i = -\alpha_i$

Since, $\alpha_i \geq 0 \implies \beta_i \leq 0$

But, $\beta_i \geq 0$

From the above two it follows that $\beta_i = 0 \; \forall i \; \in [n]$

Using the value of β_i calculated above to eliminate it from the dual, the dual is reduced to:

$$D_2' = \max_{\alpha_i \geq 0} \left(\sum_{i=1}^{n} \alpha_i - \sum_{i=1}^{n} \frac{(\alpha_i)^2}{4C} - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} (\alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle) \right)$$

which is the same as the dual derived for the problem without the additional constraint $\xi_i \geq 0$

The above can also be seen as follows:

The only contribution of β_i in the dual is in the term $-\sum_{i=1}^{n}(\alpha_i + \beta_i)^2/4C$. Since the term is negative in order to maximize the dual, this term should be minimized with respect to β_i. Now, since $\alpha_i \geq 0 \; and \; \beta_i \geq 0$, this means that $(\alpha_i + \beta_i)^2 > (\alpha_i)^2$. Thus, to maximize the dual all the β_i's must be 0.

Thus, we get that inserting the condition $\xi_i \geq 0$ does not change the optimization problem at all.

Hence Proved